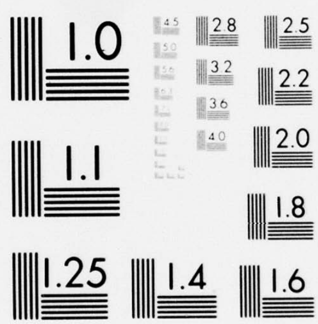
AD-A057 964 SYRACUSE UNIV NY DEPT OF ELECTRICAL AND COMPUTER EN--ETC F/G 9/5
COMPUTER PROGRAMS FOR MUTUAL COUPLING IN A FINITE PLANAR RECTAN--ETC(U)
JUL 78 J LUZWICK, R F HARRINGTON N00014-76-C-0225

UNCLASSIFIED TR-78-8 NL

| OF |

AD A057964

END
DATE FILMED
10-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

by

John Luzwick
Roger F. Harrington

Department of
Electrical and Computer Engine
Syracuse University
Syracuse, New York 13210

TECHNICAL REPORT NO. 8

July 1978

Contract No. N00014-76-C-0225

Approved for public release; distributio

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) TR-78-8	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) COMPUTER PROGRAMS FOR MUTUAL COUPLING IN A FINITE PLANAR RECTANGULAR WAVEGUIDE ANTENNA ARRAY		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Report No. 8
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) John Luzwick, Roger F. Harrington		8. CONTRACT OR GRANT NUMBER(s) (15) N00014-76-C-0225
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dept. of Electrical and Computer Engineering, Syracuse University, Syracuse, New York 13210		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (12) 51p.
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy, Office of Naval Research, Arlington, Virginia 22217		12. REPORT DATE (11) July 1978
		13. NUMBER OF PAGES 50
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Admittance matrix	Computer programs
Aperture array	Mutual coupling
Antenna array	Waveguide-fed apertures

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computer program is given to implement the equations derived in Technical Report TR-78-7, "Mutual Coupling in a Finite Planar Rectangular Waveguide Antenna Array." The aperture dimensions can be less than or equal to the feeding waveguide dimensions. Three specific uniformly spaced array lattices are considered - linear, rectangular, and isosceles triangular. For the linear and rectangular lattice cases, the Toeplitz symmetry property of the admittance matrix is utilized to save storage space and computation time.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED 406 232

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 08 22 017

20. ABSTRACT (cont.)

The program consists of a main program and several subroutines which calculate both the half-space admittance and scattering matrices. The computer program is described and listed, along with sample input-output data.

CONTENTS

I. INTRODUCTION

Computer programs for calculating the half-space admittance and scattering matrices for a finite planar rectangular waveguide antenna array are described and listed in this report. The aperture dimensions can be less than or equal to the feeding waveguide dimensions and the elements are uniformly spaced in each of two directions in either a rectangular or isosceles triangular lattice. The general theory and method of computation are given in the report [1]. Equations drawn from reference [1] are preceded by 1-. For instance, (1-5) denotes equation five of reference [1].

In summary, the procedure is an application of the method of moments to an integral equation formulation of the problem. The unknowns to be determined are the coefficients of the equivalent magnetic currents, which are proportional to the tangential electric field in the aperture regions. A single expansion function is used to approximate the electric field in each aperture.

The computer program subroutine AY which calculates the modal coefficients A_{ik}^{TE}, A_{ik}^{TM} (1-30,31) and the characteristic admittances Y_k^{TE}, Y_k^{TM} (1-34,35) is described and listed in Section II. The subroutine YHSP which calculates elements of the first column of the half-space admittance matrix Y^{hs} (1-12) is described and listed in Section III. The subroutines FCT and FCTA which supply the integrands respectively of the single integrals of (1-68) and the double integrals of (1-59) to the numerical integration subroutines QG8 and QG6A-QG6B for calculation of Y^{hs} are described and listed in Section IV. The subroutine INT which transfers data (limits of integrations and integrand specification) from subroutine YHSP to subroutine QG8 is described and listed in Section V. The subroutines QG6A-QG6B and QG8 which are respectively a six-point double and an eight-point single Gaussian quadrature numerical integration subroutine are described and listed in Section VI. The subroutine CSMTZ which solves equation (1-15) where $[Y^{wg} + Y^{hs}]$ is symmetric Toeplitz (occurs for the uniformly spaced linear array lattice case) is described and listed in Section VII. The subroutines MATMLT, TRMMLT, MULTTR,

MATVCA, and LINSLV which solve equation (1-15) where $[Y^{wg} + Y^{hs}]$ is symmetric block-Toeplitz (occurs for the uniformly spaced rectangular array lattice case) is described and listed in Section VIII. The subroutines LINEQ and DECOMP-SOLVE which are respectively a complex matrix inversion and a Gaussian elimination - LU decomposition complex linear equation solver subroutine are described and listed in Section IX. The subroutine TOPGEN generates a complete matrix (half-space admittance or scattering) given one column is described and listed in Section X. A main program which uses subroutines AY, YHSP, TOPGEN, CSMTZ, DECOMP, SOLVE, and LINSLV to obtain the half-space admittance and scattering matrices is described and listed in Section XI along with sample input-output data.

II. DESCRIPTION OF THE SUBROUTINE AY

The subroutine AY(A,Y) stores the submatrices defined by (1-30) and (1-31) in A and the admittances defined by (1-34) and (1-35) in Y,

A_k^{TE} is stored in A(k = (m+1)/2 + n/2 * LM)

Y_k^{TE} is stored in Y(k = (m+1)/2 + n/2 * LM)

m = 1,3,5,..., LM

n = 0,2,4,..., LN

A_k^{TM} is stored in A(k = LM*LN + (m+1)/2 + (n-2)/2 * LM)

Y_k^{TM} is stored in Y(k = LM*LN + (m+1)/2 + (n-2)/2 * LM)

m = 1,3,5,..., LM

n = 2,4,6,..., LN .

Note that only odd m and even n modes are calculated and that the first subscript of A_{ik}^{TE} and A_{ik}^{TM} in (1-30) and (1-31) has been dropped since the same expansion function is used in every waveguide region, $A_{1k} = A_{2k} = \ldots = A_{Nk} = A_k$. The variables AL, AL1, BL, BL1, LM, LN, ER, and ETA in the COMMON statement are respectively a/λ, a'/λ, b/λ, b'/λ, LM, LN, ε_r, and η where λ is the free space wavelength. Minimum allocations are given by

```
COMPLEX Y(2 * LM * LN + LM)
DIMENSION A(2 * LM * LN), F(LM), STE(LN),
        STM(LN).
```

DO loop 10 stores

$$\frac{b'}{a\,a'\pi\lambda^3}\sqrt{\frac{2\varepsilon_n}{aa'\,bb'}}\cos\frac{n\pi}{2}\,\frac{\sin\frac{n\pi b'}{2b}}{\frac{n\pi b'}{2b}} \quad \text{in STE } (\tfrac{n}{2}+1)$$

$$\frac{-nb'}{a'b\pi\lambda^3}\sqrt{\frac{2}{aa'\,bb'}}\cos\frac{n\pi}{2}\,\frac{\sin\frac{n\pi b'}{2b}}{\frac{n\pi b'}{2b}} \quad \text{in STM } (\tfrac{n}{2}) \; .$$

If n = 0, statement 30 calculates STE(1) using sin(0)/0 = 1. DO loop 12 stores

$$\frac{2\lambda^2}{\left(\frac{m^2}{a^2}-\frac{1}{a'^2}\right)}\sin\frac{m\pi}{2}\cos\frac{m\pi a'}{2a} \quad \text{in } F(\tfrac{m+1}{2}) \; .$$

If $|m - a/a'| < \varepsilon$ (a small number), statement 13 calculates F((m+1)/2) by replacing $(1/(m^2/a^2 - 1/a'^2))\cos(m\pi a'/2a)$ by its limit $(-\pi a^2/4)$ as m approaches a/a'. DO loop 20 calculates the coefficients A_k^{TE}, A_k^{TM}, Y_k^{TE}, and Y_k^{TM}. Statement 31 stores A_k^{TE} in A(k) where

$$A_k^{TE} = A(k) = \frac{m}{\sqrt{\left(\frac{m}{2a}\right)^2 + \left(\frac{n}{2b}\right)^2}}\,\text{STE}(\tfrac{n}{2}+1)\;F(\tfrac{m+1}{2})$$

where

$$k = (m+1)/2 + n/2 * LM \begin{cases} m = 1,3,5,\ldots, LM \\ n = 0,2,4,\ldots, LN \; . \end{cases}$$

Statement 33 stores A_k^{TM} in A(k) where

$$A_k^{TM} = A(k) = \frac{STE(\frac{n}{2})\ F(\frac{m+1}{2})}{\sqrt{(\frac{m}{2a})^2 + (\frac{n}{2b})^2}}$$

where

$$k = LM * LN + (m+1)/2 + (n-2)/2 * LM \begin{cases} m = 1,3,5,\ldots,LM \\ n = 2,4,6,\ldots,LN\ . \end{cases}$$

Statement 22 or 24 stores Y_i^{TE} of (1-34) in Y(i) where

$$i = (m+1)/2 + n/2 * LM \begin{cases} m = 1,3,5,\ldots,LM \\ n = 0,2,4,\ldots,LN \end{cases} .$$

If the calculated value of $(k_i/k)^2 - \varepsilon_r$ is zero, then statement 23 replaces $(k_i/k)^2 - \varepsilon_r$ by $10^{-6}\ \varepsilon_r$. Statement 32 stores Y_i^{TM} of (1-35) in Y(i) where

$$i = LM * LN + (m+1)/2 + (n-2)/2 * LM \begin{cases} m = 1,3,5,\ldots,LM \\ n = 2,4,6,\ldots,LN \end{cases} .$$

```
C     LISTING OF THE SUBROUTINE AY
C
      SUBROUTINE AY(A,Y)
      COMPLEX U,Y(55)
      DIMENSION A(50),F(5),STE(5),STM(5)
      COMMON AL1,BL1,PI,PI2,PI3,U/R1/ETA/R5/AL,BL
      COMMON /R6/ER,LM,LN,NT
      A1=SQRT(1./(AL*AL1*BL*BL1))
      A2=A1*BL1/(AL*AL1*PI)
      A3=1.414214*A2
      A4=1./(AL*AL)
      A5=1./(AL1*AL1)
      A6=PI3*AL1/AL
      A7=2.*AL
      B1=BL1/BL
      B2=B1*PI3
      B3=-1.414214*A1*B1/(PI*AL1)
      B4=2.*BL
      ETA2=ETA*ETA
```

```
      LN1=LN+1
   30 STE(1)=A2
      JN1=2
      JN2=2
      DO 10 JN=1,LN
      C1=JN1*B2
      C2=SIN(C1)/C1*(-1)**JN
      IF(JN.EQ.LN) GO TO 11
      STE(JN2)=A3*C2
   11 STM(JN)=JN1*B3*C2
      JN1=JN1+2
      JN2=JN2+1
   10 CONTINUE
      JM1=1
      DO 12 JM=1,LM
      JM2=JM+1
      IF(ABS(JM-AL/AL1).LT.0.001) GO TO 13
      F(JM)=2./(JM1*JM1*A4-A5)*COS(JM1*A6)*(-1)**JM2
      GO TO 14
   13 F(JM)=-PI*AL*AL/2.*(-1)**JM2
   14 JM1=JM1+2
   12 CONTINUE
      KTE=0
      KTM=LN1*LM
      KTM1=LN*LM
      JN2=0
      DO 20 JN=1,LN1
      JN1=JN-1
      BN=JN2/B4
      JM1=1
      DO 21 JM=1,LN
      AM=JM1/A7
      C1=AM*AM+BN*BN
      C2=SQRT(C1)
      C1=C1-ER
      KTE=KTE+1
      IF(C1) 22,23,24
   22 Y(KTE)=SQRT(-C1)/ETA
      GO TO 25
   23 C1=1.E-6*ER
   24 Y(KTE)=-U*SQRT(C1)/ETA
   25 IF(JN.EQ.LN1) GO TO 26
   31 A(KTE)=JM1*STE(JN)*F(JM)/C2
   26 IF(JN.EQ.1) GO TO 27
      KTM=KTM+1
      KTM1=KTM1+1
   32 Y(KTM)=ER/(Y(KTE)*ETA2)
   33 A(KTM1)=STM(JN1)*F(JM)/C2
   27 JM1=JM1+2
   21 CONTINUE
      JN2=JN2+2
   20 CONTINUE
      RETURN
      END
```

III. DESCRIPTION OF THE SUBROUTINE YHSP

The subroutine YHSP (N,X,Y,YSP) uses the single integrals of (1-68) and the double integrals of (1-59) to calculate elements of the first column of the Y^{hs} matrix (1-12). There are N apertures. X(I) and Y(I) are respectively the x and y center coordinates of the Ith aperture.

Minimum allocations are given by

```
COMPLEX YSP(N)
DIMENSION X(N), Y(N) .
```

The constants D(1) - D(6) are the multiplying factors $K_2 - K_5$ and $K_7 - K_8$ (1-61 to 64, 1-66 to 67) of the integrands of (1-59),

$$D(1) = \lambda\{(1 - 1/4a'^2)\ (x_i - x_j + a')\cos\frac{\pi}{a'}(x_j - x_i) + \frac{a'}{\pi}(1 + 1/4a'^2)\sin\frac{\pi}{a'}(x_j - x_i)\}$$

$$D(2) = \lambda\{(1 - 1/4a'^2)\ (x_i - x_j + a')\sin\frac{\pi}{a'}(x_j - x_i) - \frac{a'}{\pi}(1 + 1/4a'^2)\cos\frac{\pi}{a'}(x_j - x_i)\}$$

$$D(3) = (1 - 1/4a'^2)\cos\frac{\pi}{a'}(x_j - x_i)$$

$$D(4) = (1 - 1/4a'^2)\sin\frac{\pi}{a'}(x_j - x_i)$$

$$D(5) = \lambda\{(1 - 1/4a'^2)\ (x_j - x_i + a')\cos\frac{\pi}{a'}(x_j - x_i) - \frac{a'}{\pi}(1 + 1/4a'^2)\sin\frac{\pi}{a'}(x_j - x_i)\}$$

$$D(6) = \lambda\{(1 - 1/4a'^2)\ (x_j - x_i + a')\sin\frac{\pi}{a'}(x_j - x_i) + \frac{a'}{\pi}(1 + 1/4a'^2)\cos\frac{\pi}{a'}(x_j - x_i)\}$$

(a', x_i, and x_j are distances per unit wavelength).

The constants TH1 - TH9 are angles at which a rotating line from the origin (line O-A in Fig. 1) will intersect different straight line segment junctions defined by $x = x_j - x_i - a'$, $x_j - x_i$, $x_j - x_i + a'$, and $y = y_j - y_i - b'$, $y_j - y_i$, $y_j - y_i + b'$ (see Fig. 1).

The CALL INT(N4, N5, N6, P1, P2, XL, XU) statements between 12 and 11 select the correct $\rho_1(\theta)$ and $\rho_2(\theta)$ expressions used in $M_1(\theta) - M_6(\theta)$ (1-77 to 82), select the integrand for each subarea (I-IV) in which the single integration is being performed, and supply the θ limits of integration.

For

N4 = 0,	$\rho_1(\theta) = 0$
N4 = 1,	$\rho_1(\theta) = P1/\cos\theta$
N4 = 2,	$\rho_1(\theta) = P1/\sin\theta$
N5 = 0,	$\rho_2(\theta) = 0$
N5 = 1,	$\rho_2(\theta) = P2/\cos\theta$
N5 = 2,	$\rho_2(\theta) = P2/\sin\theta$
N6 < 2,	Integration over a subarea I-IV is still being performed
N6 = 2,	Multiplying factors for subareas I and III are used
N6 = 3,	Multiplying factors for subareas II and IV are used

where P1 and P2 are lines which define the subareas of integration (I-IV) and, therefore, represent the limits of integration for the variable ρ. For instance, in Fig. 1, the lines which define the limits of integration along line OA in subarea IV are $P1 = y_j - y_i$ and $P2 = y_j - y_i + b'$. XL and XU represent respectively the θ_{lower} and θ_{upper} limits of integration.

The arguments of the CALL INT statements will vary since the area of integration (I-IV) can move throughout the first quadrant of the uv plane (assumes $x_i \leq x_j$). There will be overlapping into other quadrants

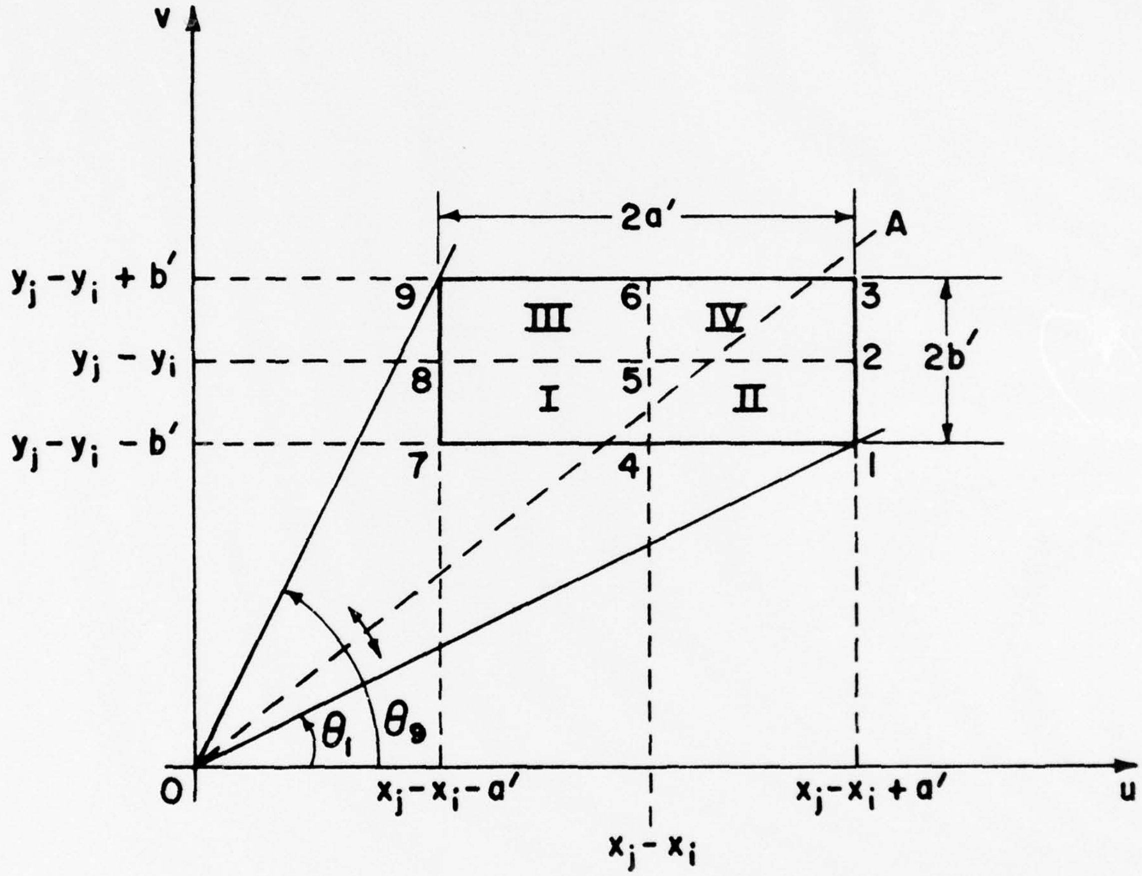


Fig. 1. Integration subareas for solution involving the single integrals of (1-68).

when $x_i = x_j$, $y_i \neq y_j$, $x_i \neq x_j$, $y_i = y_j$, and $x_i = x_j$, $y_i = y_j$. There are four specific cases to consider for determining the integrations to be performed:

Case 1) $x_i \neq x_j$, $y_i \neq y_j$ - this integration will require twelve CALL INT statements representing the twelve integration areas of subareas (I-IV). The twelve statements are located between statements 13 and 14.

Case 2) $x_i = x_j$, $y_i \neq y_j$ - this integration will require only four CALL INT statements because of symmetry considerations. The four statements are located between statements 14 and 15.

Case 3) $x_i \neq x_j$, $y_i = y_j$ - this integration like case 2 will require only four statements which are located between statements 15 and 11.

Case 4) $x_i = x_j$, $y_i = y_j$ - this integration which represents the calculation of the self-admittance will require only two CALL INT statements which are located between statements 12 and 13.

If $(x_j - x_i) > 4a$, a solution involving the double integrals of (1-59) is used. The calling statement is CALL QG6A(XL,XU,YL,YU,N9,Z). The variables XL, XU, YL, YU represent respectively the lower and upper limits of integration for the u and v variables of integration in (1-59).

If N9 = 1, the first of the four double integrals of (1-59) are evaluated (if N9 = 2, the second, etc.). There are three specific cases to consider for determining the integrations to be performed:

Case 1) $x_i \neq x_j$, $y_i \neq y_j$ - this integration will require four CALL QG6A statements representing the four subareas of integration (I-IV). The four statements are located between statements 11 and 25.

Case 2) $x_i = x_j$, $y_i \neq y_j$ - this integration will require only two CALL QG6A statements because of symmetry considerations. The two statements are located between statements 25 and 26.

Case 3) $x_i \neq x_j$, $y_i = y_j$ - this integration like case 2 will require only two CALL QG6A statements which are located between statements 26 and 10.

```
C      LISTING OF THE SUBROUTINE YHSP
C
       SUBROUTINE YHSP(N.X.Y.YSP)
       COMPLEX G3.G5.G7.G9.S(10).SA.SB.U.U1.YSP(6)
       DIMENSION X(6).Y(6)
       COMMON A1.B1.PI.PI2.PI3.U/R1/ETA/R2/D(6).S
       COMMON /R3/G1.G2.G3.G4.G5.G6.G7.G8.G9.G10.G11
       COMMON /R5/AL.BL/R7/YY3.YY4
       U1=U/(A1*B1*ETA)
       DO 10 J=1.N
       X1=X(1)
       X2=X(J)
       Y1=Y(1)
       Y2=Y(J)
       XX1=X2-X1-A1
       XX2=X2-X1
       XX3=X2-X1+A1
       XX4=X1-X2+A1
       YY1=Y2-Y1-B1
       YY2=Y2-Y1
       YY3=Y2-Y1+B1
       YY4=Y1-Y2+B1
       XA=(X2-X1)/G1
       DI=4.*AL
       C1=COS(XA)
       C2=SIN(XA)
       D(3)=C1*G10
       D(4)=C2*G10
       C1=C1*G11
       C2=C2*G11
       D(1)=XX4*D(3)+C2
       D(2)=XX4*D(4)-C1
       D(5)=XX3*D(3)-C2
       D(6)=XX3*D(4)+C1
       IF(XX2.GT.DI) GO TO 11
       TH1=ATAN2(YY1.XX3)
       TH2=ATAN2(YY2.XX3)
       TH3=ATAN2(YY3.XX3)
       TH4=ATAN2(YY1.XX2)
       TH6=ATAN2(YY3.XX2)
       TH7=ATAN2(YY1.XX1)
```

```
      TH8=ATAN2(YY2,XX1)
      TH9=ATAN2(YY3,XX1)
      IF(J.EQ.1) GO TO 12
      TH5=ATAN2(YY2,XX2)
   12 IF(J.GT.1) GO TO 13
      CALL INT(0,1,0,0.,XX3,0.,TH3)
      CALL INT(0,2,3,0.,YY3,TH3,PI3)
      YSP(J)=4.*(YY3*S(9)-S(10))
      GO TO 10
   13 IF((X1.EQ.X2).AND.(Y1.NE.Y2)) GO TO 14
      IF((X1.NE.X2).AND.(Y1.EQ.Y2)) GO TO 15
      IF(TH7.GT.TH5) GO TO 16
      CALL INT(2,1,0,YY1,XX2,TH4,TH7)
      CALL INT(1,1,1,XX1,XX2,TH7,TH5)
      CALL INT(1,2,2,XX1,YY2,TH5,TH8)
      GO TO 17
   16 CALL INT(2,1,0,YY1,XX2,TH4,TH5)
      CALL INT(2,2,1,YY1,YY2,TH5,TH7)
      CALL INT(1,2,2,XX1,YY2,TH7,TH8)
   17 SA=YY4*S(9)
      SB=S(10)
      IF(TH8.GT.TH6) GO TO 18
      CALL INT(2,1,0,YY2,XX2,TH5,TH8)
      CALL INT(1,1,1,XX1,XX2,TH8,TH6)
      CALL INT(1,2,2,XX1,YY3,TH6,TH9)
      GO TO 19
   18 CALL INT(2,1,0,YY2,XX2,TH5,TH6)
      CALL INT(2,2,1,YY2,YY3,TH6,TH8)
      CALL INT(1,2,2,XX1,YY3,TH8,TH9)
   19 SA=SA+YY3*S(9)
      SB=SB-S(10)
      YSP(J)=SA+SB
      IF(TH4.GT.TH2) GO TO 20
      CALL INT(2,1,0,YY1,XX3,TH1,TH4)
      CALL INT(1,1,1,XX2,XX3,TH4,TH2)
      CALL INT(1,2,3,XX2,YY2,TH2,TH5)
      GO TO 21
   20 CALL INT(2,1,0,YY1,XX3,TH1,TH2)
      CALL INT(2,2,1,YY1,YY2,TH2,TH4)
      CALL INT(1,2,3,XX2,YY2,TH4,TH5)
   21 SA=YY4*S(9)
      SB=S(10)
      IF(TH5.GT.TH3) GO TO 22
      CALL INT(2,1,0,YY2,XX3,TH2,TH5)
      CALL INT(1,1,1,XX2,XX3,TH5,TH3)
      CALL INT(1,2,3,XX2,YY3,TH3,TH6)
      GO TO 23
   22 CALL INT(2,1,0,YY2,XX3,TH2,TH3)
      CALL INT(2,2,1,YY2,YY3,TH3,TH5)
      CALL INT(1,2,3,XX2,YY3,TH5,TH6)
   23 SA=SA+YY3*S(9)
      SB=SB-S(10)
      YSP(J)=YSP(J)+SA+SB
      GO TO 10
   14 CALL INT(2,2,0,YY1,YY2,PI3,TH8)
      CALL INT(2,1,2,YY1,XX1,TH8,TH7)
      SA=YY4*S(9)
      SB=S(10)
```

```
      CALL INT(2.2.0.YY2.YY3.PI3.TH9)
      CALL INT(2.1.2.YY2.XX1.TH9.TH8)
      SA=SA+YY3*S(9)
      SB=SB-S(10)
      YSP(J)=2.*(SA+SB)
      GO TO 10
   15 CALL INT(1.2.0.XX1.YY1.TH7.TH4)
      CALL INT(1.1.2.XX1.XX2.TH4.0.)
      YSP(J)=YY4*S(9)+S(10)
      CALL INT(1.2.0.XX2.YY1.TH4.TH1)
      CALL INT(1.1.3.XX2.XX3.TH1.0.)
      YSP(J)=2.*(YSP(J)+YY4*S(9)+S(10))
      GO TO 10
   11 IF((X1.EQ.X2).AND.(Y1.NE.Y2)) GO TO 25
      IF((X1.NE.X2).AND.(Y1.EQ.Y2)) GO TO 26
      CALL QG6A(XX1.XX2.YY1.YY2.1.SA)
      SB=SA
      CALL QG6A(XX2.XX3.YY1.YY2.2.SA)
      SB=SB+SA
      CALL QG6A(XX1.XX2.YY2.YY3.3.SA)
      SB=SB+SA
      CALL QG6A(XX2.XX3.YY2.YY3.4.SA)
      YSP(J)=SA+SB
      GO TO 10
   25 CALL QG6A(XX1.XX2.YY1.YY2.1.SA)
      SB=2.*SA
      CALL QG6A(XX1.XX2.YY2.YY3.3.SA)
      YSP(J)=2.*SA+SB
      GO TO 10
   26 CALL QG6A(XX1.XX2.YY1.YY2.1.SA)
      SB=2.*SA
      CALL QG6A(XX2.XX3.YY1.YY2.2.SA)
      YSP(J)=2.*SA+SB
   10 CONTINUE
      DO 30 I=1.N
      YSP(I)=U1*YSP(I)
   30 CONTINUE
      RETURN
      END
```

IV. DESCRIPTION OF THE SUBROUTINES FCT AND FCTA

The subroutine FCT (W,X) supplies the integrands for the single numerical integration subroutine QG8. The W(I) are the integrands of equation (1-68) evaluated at $\theta = X$ where X is the variable of integration for QG8.

The logic between statements 40 and 6 determines the $\rho_1(\theta)$ and $\rho_2(\theta)$ expressions used in $M_1(\theta) - M_6(\theta)$ (1-77 to 82).

For

$$\begin{array}{ll}
N4 = 0\ , & C = \rho_1(\theta) = 0 \\
N4 = 1\ , & C = \rho_1(\theta) = P1/\cos\theta \\
N4 = 2\ , & C = \rho_1(\theta) = P1/\sin\theta \\
N5 = 0\ , & D = \rho_2(\theta) = 0 \\
N5 = 1\ , & D = \rho_2(\theta) = P2/\cos\theta \\
N5 = 2, & D = \rho_2(\theta) = P2/\sin\theta .
\end{array}$$

W(1) - W(8) represent respectively $L_1(\theta) - L_8(\theta)$ (1-69 to 76).

The logic statements between 41 and 37 supply the limiting expressions (1-84 to 99) when $\left|\frac{\pi}{a'}\cos\theta \pm k\right| < \varepsilon$ (a small value). For a first order zero $(\frac{\pi}{a'}\cos\theta \pm k)$ in the denominator of $L_1(\theta) - L_8(\theta)$ (1-69 to 76), the ε value of E1 = 0.0002 was used while for higher order zeros, ε = E2 = 0.002 was used.

The subroutine FCTA(N9,X,Y,Z) supplies the integrands for the double numerical integration subroutines QG6A and QG6B. N9 specifies which of the four double integrals of (1-59) will be evaluated (if N9 = 1, the first of the four, if N9 = 2, the second, etc.). X and Y are the variables of integration respectively for QG6A and QG6B. Z is the integrand of one of the four double integrals of (1-59) evaluated at X and Y. The integrands of the double integrals of (1-59) are transferred from subroutine YHSP to subroutine FCTA through the COMMON/R2/ and COMMON/R7/ statements.

```
C     LISTINGS OF THE SUBROUTINES FCT AND FCTA
C
      SUBROUTINE FCT(W,X)
      COMPLEX E(8),F(8),G3,G5,G7,G9,SA,SB,U,W(8)
      COMMON A1,B1,PI,PI2,PI3,U/R3/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
      COMMON /R4/N4,N5,N6,P1,P2,XL,XU
      C1=COS(X)
      C2=SIN(X)
      E1=0.0002
      E2=0.002
      P=C1/A1
      S=PI*(P-2.)
      T=PI*(P+2.)
   40 IF(N4.NE.0) GO TO 1
      C=0.
      GO TO 3
    1 IF(N4.EQ.2) GO TO 2
      C=P1/C1
      GO TO 3
    2 C=P1/C2
    3 IF(N5.NE.0) GO TO 4
      D=0.
      GO TO 6
    4 IF(N5.EQ.2) GO TO 5
      D=P2/C1
      GO TO 6
    5 D=P2/C2
    6 Y=C1-2.*A1
      P=D*S
      E(1)=COS(P)+U*SIN(P)
      P=C*S
      E(2)=COS(P)+U*SIN(P)
      Z=C1+2.*A1
      P=D*T
      F(1)=COS(P)-U*SIN(P)
      P=C*T
      F(2)=COS(P)-U*SIN(P)
      E(3)=(E(1)-E(2))/Y
      F(3)=(F(1)-F(2))/Z
      SA=D*E(1)
      SB=C*E(2)
      E(4)=(SA-SB)/Y
      SA=D*SA
      SB=C*SB
      E(6)=(SA-SB)/Y
      SA=D*F(1)
      SB=C*F(2)
      F(4)=(SA-SB)/Z
      E(5)=E(3)/Y
      F(5)=F(3)/Z
      SA=D*SA
      SB=C*SB
      F(6)=(SA-SB)/Z
      E(7)=E(4)/Y
      F(7)=F(4)/Z
      E(8)=E(5)/Y
      F(8)=F(5)/Z
   41 IF(ABS(Y).LT.E1) GO TO 20
      IF(ABS(Z).LT.E1) GO TO 21
      W(1)=G3*(E(3)-F(3))
```

```
      GO TO 22
   20 W(1)=-G3*F(3)+0.5*(D-C)
      GO TO 22
   21 W(1)=G3*E(3)+0.5*(D-C)
   22 IF(ABS(Y).LT.E1) GO TO 23
      IF(ABS(Z).LT.E1) GO TO 24
      W(2)=-G2*(E(3)+F(3))
      GO TO 25
   23 W(2)=-G2*F(3)-0.5*U*(D-C)
      GO TO 25
   24 W(2)=-G2*E(3)+0.5*U*(D-C)
   25 IF(ABS(Y).LT.E2) GO TO 26
      IF(ABS(Z).LT.E2) GO TO 27
      W(5)=G3*(E(4)-F(4))+G6*(E(5)+F(5))
      GO TO 28
   26 H=0.25*(D*D-C*C)
      W(5)=-G3*F(4)+G6*F(5)+H
      GO TO 28
   27 H=0.25*(D*D-C*C)
      W(5)=G3*E(4)+G6*E(5)+H
   28 W(3)=C1*W(5)
      W(5)=C2*W(5)
      IF(ABS(Y).LT.E2) GO TO 29
      IF(ABS(Z).LT.E2) GO TO 30
      W(6)=-G2*(E(4)+F(4))+G7*(E(5)-F(5))
      GO TO 31
   29 W(6)=-G2*F(4)-G7*F(5)-U*H
      GO TO 31
   30 W(6)=-G2*E(4)+G7*E(5)+U*H
   31 W(4)=C1*W(6)
      W(6)=C2*W(6)
      C1=C1*C2
      IF(ABS(Y).LT.E2) GO TO 32
      IF(ABS(Z).LT.E2) GO TO 33
      W(7)=C1*(G3*(E(6)-F(6))+G4*(E(7)+F(7))-G9*(E(8)-F(8)))
      GO TO 34
   32 H=(D*D*D-C*C*C)/6.
      W(7)=C1*(-G3*F(6)+G4*F(7)+G9*F(8)+H)
      GO TO 34
   33 H=(D*D*D-C*C*C)/6.
      W(7)=C1*(G3*E(6)+G4*E(7)-G9*E(8)+H)
   34 IF(ABS(Y).LT.E2) GO TO 35
      IF(ABS(Z).LT.E2) GO TO 36
      W(8)=C1*(-G2*(E(6)+F(6))+G5*(E(7)-F(7))+G8*(E(8)+F(8)))
      GO TO 37
   35 W(8)=C1*(-G2*F(6)-G5*F(7)+G8*F(8)-U*H)
      GO TO 37
   36 W(8)=C1*(-G2*E(6)+G5*E(7)+G8*E(8)+U*H)
   37 CONTINUE
      RETURN
      END
      SUBROUTINE FCTA(N9,X,Y,Z)
      COMPLEX G,G3,G5,G7,G9,S(10),U,Z
      COMMON AL1,BL1,PI,PI2,PI3,U/R2/D(6),S
      COMMCN /R3/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11/R7/YY3,YY4
      P1=X/G1
      C1=COS(P1)
      C2=SIN(P1)
      C3=SQRT(X*X+Y*Y)
      C4=PI2*C3
```

```
      G=(COS(C4)-U*SIN(C4))/C3
      IF(N9.EQ.2) GO TO 1
      IF(N9.EQ.3) GO TO 2
      IF(N9.EQ.4) GO TO 3
      Z=G*(YY4+Y)*(C1*(D(1)+X*D(3))+C2*(D(2)+X*D(4)))
      RETURN
    1 Z=G*(YY4+Y)*(C1*(D(5)-X*D(3))+C2*(D(6)-X*D(4)))
      RETURN
    2 Z=G*(YY3-Y)*(C1*(D(1)+X*D(3))+C2*(D(2)+X*D(4)))
      RETURN
    3 Z=G*(YY3-Y)*(C1*(D(5)-X*D(3))+C2*(D(6)-X*D(4)))
      RETURN
      END
```

V. DESCRIPTION OF THE SUBROUTINE INT

The subroutine INT(N4, N5, N6, P1, P2, XL, XU) transfers the data N4, N5, N6 which is supplied by the subroutine YHSP and required by the subroutine FCT through the COMMON/R4/ statement, calls the single numerical integration subroutine QG8, and calculates the S(9) and S(10) terms which represent the four single integrals of (1-68). The arguments N4, N5, N6, P1, P2, XL, and XU are defined in the description of the subroutine YHSP and will not be repeated here.

The following logical statements are located between statements 3 and 4. If N6 is less than 2, no S(9) or S(10) term is calculated (one or two more intermediate integrations are required before evaluation). Also, if N6 = 0, the variable S which represents the result of intermediate integrand evaluations of subroutine QG8 is reset to zero. If N6 = 2, S(9) represents the first single integral (I) of (1-68) and S(10) represents the third (III). If N6 = 3, S(9) represents the second integral (II) of (1-68) and S(10) represents the fourth (IV).

```
C     LISTING OF THE SUBROUTINE INT
C
      SUBROUTINE INT(N4,N5,N6,P1,P2,XL,XU)
      COMPLEX S(10),T(8)
      COMMON /R2/D(6),S/R4/N7,N8,N9,P3,P4,XL1,XU1
      N7=N4
      N8=N5
      N9=N6
      P3=P1
      P4=P2
      XL1=XL
      XU1=XU
      IF(N6.NE.0) GO TO 1
      DO 2 I=1,8
      S(I)=(0.,0.)
    2 CONTINUE
    1 CALL QG8(T)
      DO 3 I=1,8
      S(I)=S(I)+T(I)
    3 CONTINUE
      IF(N6.LT.2) GO TO 4
      IF(N6.EQ.2) S(9)=D(1)*S(1)+D(2)*S(2)+D(3)*S(3)+D(4)*S(4)
      IF(N6.EQ.3) S(9)=D(5)*S(1)+D(6)*S(2)-D(3)*S(3)-D(4)*S(4)
      IF(N6.EQ.2) S(10)=D(1)*S(5)+D(2)*S(6)+D(3)*S(7)+D(4)*S(8)
      IF(N6.EQ.3) S(10)=D(5)*S(5)+D(6)*S(6)-D(3)*S(7)-D(4)*S(8)
    4 CONTINUE
      RETURN
      END
```

VI. DESCRIPTION OF THE SUBROUTINES QG6A, QG6B, AND QG8

The combination of subroutines QG6A(XL, XU, YL, YU, N9, Z) and QG6B(X, YL, YU, N9, Z) perform a six-point Gaussian quadrature double numerical integration on the double integrals of (1-59). The basic integral under consideration can be represented as

$$I = \int_{XL}^{XU} dx \int_{YL}^{YU} dy\ F(x,y)\ . \tag{1}$$

This integral can be transformed from the integration intervals (XL, XU), (YL, YU) to the intervals (-1, 1), (-1, 1) by the transformation

$$x' = \frac{2x - (XU + XL)}{(XU - XL)} \tag{2}$$

$$y' = \frac{2y - (YU + YL)}{(YU - YL)} \tag{3}$$

so that (1) becomes

$$I = \left(\frac{XU-XL}{2}\right)\left(\frac{YU-YL}{2}\right) \int_{-1}^{1} dx' \int_{-1}^{1} dy'\ F\left(\left[\frac{XU-XL}{2}\ x' + \frac{XU+XL}{2}\right], \left[\frac{YU-YL}{2}\ y' + \frac{YU+YL}{2}\right]\right). \tag{4}$$

A double six-point Gaussian quadrature formula [2, Section 7.2] is used to evaluate (4),

$$I = \left(\frac{XU-XL}{2}\right)\left(\frac{YU-YL}{2}\right) \sum_{i=1}^{6} \sum_{j=1}^{6} A_i^{(6)} A_j^{(6)}\ F(x_i^{(6)},\ y_j^{(6)}) \tag{5}$$

where $x_i^{(6)}$, $y_j^{(6)}$ are the roots of the Legendre polynomial of degree 6 - $P_6(x_i^{(6)}$ or $y_j^{(6)}) = 0$ (the Legendre polynomials are orthogonal on the (-1, 1) interval). The coefficients $A_i^{(6)}$, $A_j^{(6)}$ are defined as

$$A_i^{(6)} = \frac{1}{3\ P_6'\ (x_i^{(6)})\ P_5(x_i^{(6)})} \tag{6}$$

$$A_j^{(6)} = \frac{1}{3\ P_6'\ (y_j^{(6)})\ P_5(y_j^{(6)})} \tag{7}$$

(' represents derivative) and are tabulated [2, p. 337].

In the calling arguments, (XL, XU, YL, YU) represents respectively the lower and upper limits of integration for the x and y variables, N9 is used in the subroutine FCTA to specify the function to be integrated (if N9 = 1, the first double integral integrand of (1-59) is specified, if N9 = 2, the second, etc.) and Z is the result of the integration.

Subroutine QG8(Y) performs a single eight-point Gaussian quadrature numerical integration on the single integrals of (1-68). In the calling argument, Y is the result of the integration. The variables N4, N5, N6, P1, P2, XL, and XU in the COMMON/R4/ statement are defined in the description of subroutine YHSP and will not be repeated here.

```
C     LISTINGS OF THE SUBROUTINE QG6A,QG6B,AND QG8
C
      SUBROUTINE QG6A(XL,XU,YL,YU,N9,Z)
      COMPLEX Z,Z1,Z2
      A=0.5*(XL+XU)
      B=XU-XL
      C=0.4662348*B
      X=A+C
      CALL QG6B(X,YL,YU,N9,Z1)
      X=A-C
      CALL QG6B(X,YL,YU,N9,Z2)
      Z=0.08566225*(Z1+Z2)
      C=0.3306047*B
      X=A+C
      CALL QG6B(X,YL,YU,N9,Z1)
      X=A-C
      CALL QG6B(X,YL,YU,N9,Z2)
      Z=Z+0.1803808*(Z1+Z2)
      C=0.1193096*B
      X=A+C
      CALL QG6B(X,YL,YU,N9,Z1)
      X=A-C
      CALL QG6B(X,YL,YU,N9,Z2)
      Z=B*(Z+0.2339570*(Z1+Z2))
      RETURN
      END
      SUBROUTINE QG6B(X,YL,YU,N9,Z)
      COMPLEX Z,Z1,Z2
      A=0.5*(YL+YU)
      B=YU-YL
      C=0.4662348*B
      Y=A+C
      CALL FCTA(N9,X,Y,Z1)
```

```
      Y=A-C
      CALL FCTA(N9,X,Y,Z2)
      Z=0.08566225*(Z1+Z2)
      C=0.3306047*B
      Y=A+C
      CALL FCTA(N9,X,Y,Z1)
      Y=A-C
      CALL FCTA(N9,X,Y,Z2)
      Z=Z+0.1803808*(Z1+Z2)
      C=0.1193096*B
      Y=A+C
      CALL FCTA(N9,X,Y,Z1)
      Y=A-C
      CALL FCTA(N9,X,Y,Z2)
      Z=B*(Z+0.2339570*(Z1+Z2))
      RETURN
      END
      SUBROUTINE QG8(Y)
      COMPLEX T(8),T1(8),Y(8)
      COMMON /R4/N4,N5,N6,P1,P2,XL,XU
      A=.5*(XL+XU)
      B=XU-XL
      C=.4801449*B
      X=A+C
      CALL FCT(T,X)
      DO 1 I=1,8
      T1(I)=T(I)
    1 CONTINUE
      X=A-C
      CALL FCT(T,X)
      DO 2 I=1,8
      Y(I)=.05061427*(T(I)+T1(I))
    2 CONTINUE
      C=.3983332*B
      X=A+C
      CALL FCT(T,X)
      DO 3 I=1,8
      T1(I)=T(I)
    3 CONTINUE
      X=A-C
      CALL FCT(T,X)
      DO 4 I=1,8
      Y(I)=Y(I)+.1111905*(T(I)+T1(I))
    4 CONTINUE
      C=.2627662*B
      X=A+C
      CALL FCT(T,X)
      DO 5 I=1,8
      T1(I)=T(I)
    5 CONTINUE
      X=A-C
      CALL FCT(T,X)
      DO 6 I=1,8
      Y(I)=Y(I)+.1568533*(T(I)+T1(I))
    6 CONTINUE
      C=.09171732*B
      X=A+C
      CALL FCT(T,X)
```

```
      DO 7 I=1,8
      T1(I)=T(I)
    7 CONTINUE
      X=A-C
      CALL FCT(T,X)
      DO 8 I=1,8
      Y(I)=B*(Y(I)+.1813419*(T(I)+T1(I)))
    8 CONTINUE
      RETURN
      END
```

VII. DESCRIPTION OF THE SUBROUTINE CSMTZ

The subroutine CSMTZ(N, A, B, X) solves the set of equations

$$L_N s_N = d_N \tag{8}$$

where L_N is an $N \times N$ complex symmetric Toeplitz matrix. In the argument of CSMTZ, N is the number of unknowns in (8), A is the input array whose elements are the first row of L_N, B is the input array whose elements are those of d_N, and X is the output array whose elements are those of s_N. It is assumed that (8) is normalized so that the first element of the first row of L_N is equal to unity.

This subroutine solves equation (1-15)

$$[Y^{wg} + Y^{hs}]\vec{V} = \vec{I}^{imp} \tag{1-15}$$

where $[Y^{wg} + Y^{hs}]$ is a symmetric Toeplitz admittance matrix (linear array lattice case).

The two main advantages of the algorithm used in this subroutine are

(1) It solves equation (1-15) directly without inversion of $[Y^{wg} + Y^{hs}]$ requiring roughly $2N^2$ multiplications and divisions.

(2) It requires only the first now of $[Y^{wg} + Y^{hs}]$ and, therefore, minimizes the storage requirement for the subroutine.

The derivation of the algorithm can be found in [3, 4] and will not be repeated here. However, the algorithm logic will be presented using the notation developed in [3, 4]. The following notation will be used (the same used by Zohar in [3, 4]): Greek letters are used for scalars, capital letters for square matrices, lower case letters for column matrices, ~ denotes transpose, and ^ is a reversal symbol - $\hat{g}_k$ denotes the reversed order of g_k, that is, $(\hat{g}_k)_{i1} = (g_k)_{k+1-i,1}$.

The matrix L_N of (8) is bordered as follows,

$$L_N = \begin{bmatrix} 1 & \tilde{r}_{N-1} \\ & \\ r_{N-1} & L_{N-1} \end{bmatrix} \tag{9}$$

where $\tilde{r}_{N-1}$ = [A(2), A(3),..., A(N)] (A(i) indicates the ith component (numbered columnwise) of array A). In (8), $\tilde{d}_N = [\delta_1, \delta_2, \ldots, \delta_N]$. The algorithm is based on a recursion relation with initial values given by

$$s_1 = \delta_1 \ , \quad \rho_1 = - A(2) \ , \quad \lambda_1 = 1 - A(2) * A(2) \ . \tag{10}$$

Recursion of s_i, $\hat{e}_i$, and λ_i for i-1,2,...,N-2 is given by

$$\theta_i = \delta_{i+1} - \tilde{s}_i \hat{r}_i \tag{11}$$

$$\eta_i = - A(i+2) - \tilde{r}_i \hat{e}_i \tag{12}$$

$$s_{i+1} = \begin{bmatrix} s_i + (\theta_i/\lambda_i) \hat{e}_i \\ \\ \theta_i/\lambda_i \end{bmatrix} \tag{13}$$

$$e_{i+1} = \begin{bmatrix} e_i + (\eta_i/\lambda_i) \hat{e}_i \\ \\ \eta_i/\lambda_i \end{bmatrix} \tag{14}$$

$$\lambda_{i+1} = \lambda_i - \eta_i^2/\lambda_i \ . \tag{15}$$

The last computed values are θ_{N-1}, η_{N-2}, s_N, e_{N-1}, and λ_{N-1}. Note that $\tilde{a}_N$, g_N, and γ appearing in [3] are not needed because of symmetry.

Minimum allocations are given by

COMPLEX A(N), B(N), E(N), ES(N-1),X(N).

The initial values for recursion are first computed: $s_1 = X(1)$, $\rho_1 = E(1)$, and $\lambda_1 = LA$. DO loop 10 is the main recursion loop whose index is $i(i=1,2,\ldots,N-2)$. DO loop 11 computes $S1 = \tilde{s}_i \hat{r}_i$ and $E1 = \tilde{r}_i \hat{e}_i$. Following statement 11, η_i and θ_i are computed. DO loop 12 computes s_{i+1} and stores e_i in dummy array ES because e_i is needed in DO loop 13 to recompute itself. DO loop 13 computes e_i. Following 13, λ_{i+1} is computed and the index i of DO loop 10 is stopped. DO loops 14 and 15 compute s_N.

```
C      LISTING OF THE SUBROUTINE CSMTZ
C
       SUBROUTINE CSMTZ(N,A,B,X)
       COMPLEX A(6),B(6),E(6),ES(5),X(6)
       COMPLEX E1,ET,ET1,LA,S1,TH,TH1
       X(1)=B(1)
       E(1)=-A(2)
       LA=1.-A(2)*A(2)
       N1=N-2
       N2=N-1
       DO 10 I=1,N1
       E1=(0.,0.)
       S1=(0.,0.)
       DO 11 J=1,I
       S1=S1+X(J)*A(I-J+2)
       E1=E1+E(I-J+1)*A(J+1)
    11 CONTINUE
       ET=-A(I+2)-E1
       TH=B(I+1)-S1
       TH1=TH/LA
       ET1=ET/LA
       DO 12 K=1,I
       X(K)=X(K)+TH1*E(I-K+1)
       ES(K)=E(K)
    12 CONTINUE
       DO 13 K=1,I
       E(K)=ES(K)+ET1*ES(I-K+1)
    13 CONTINUE
       X(I+1)=TH/LA
       E(I+1)=ET1
       LA=LA-ET*ET1
    10 CONTINUE
       S1=(0.,0.)
       DO 14 J=1,N2
       S1=S1+X(J)*A(N-J+1)
    14 CONTINUE
       TH=B(N)-S1
       TH1=TH/LA
       DO 15 K=1,N2
       X(K)=X(K)+TH1*E(N-K)
    15 CONTINUE
       X(N2+1)=TH1
       RETURN
       END
```

VIII. DESCRIPTION OF THE SUBROUTINES MATMLT, TRMMLT, MULTTR, MATVCA, AND LINSLV

The subroutines MATMLT, TRMMLT, MULTTR, MATVCA, LINSLV, and the following recursion relationships are taken from a research report by D. H. Sinnott [5]. These subroutines represent an efficient algorithm for solving equation (1-15)

$$[Y^{wg} + Y^{hs}]\vec{V} = \vec{I}^{imp} \qquad (1\text{-}15)$$

where $[Y^{wg} + Y^{hs}]$ is a symmetric block-Toeplitz admittance matrix. The two main advantages of this algorithm are:

(1) It solves equation (1-15) directly without inversion of $[Y^{wg} + Y^{hs}]$ and is, therefore, more efficient.

(2) Storage requirements are considerably less than that required for an inversion solution. Approximately one quarter of the matrix $[Y^{wg} + Y^{hs}]^{-1}$ must be stored for an inversion solution while only the first row of blocks of $[Y^{wg} + Y^{hs}]$ is required for the given algorithm solution.

Recall the form of the block-Toeplitz matrix Y,

$$[Y] = Y^{(n)} = \begin{bmatrix} Y_0 & Y_1 & \cdots & Y_n \\ Y_1 & Y_0 & \cdots & Y_{n-1} \\ \vdots & & & \\ Y_n & Y_{n-1} & \cdots & Y_0 \end{bmatrix} \qquad (1\text{-}108)$$

where Y_0 is the submatrix which defines the self-admittance of an element of the array (specified number of apertures - see Fig. 1-5) and $Y_{|i-j|}$, $i \neq j$, is the submatrix which defines the mutual admittance between elements i and j of the array. The parenthesized superscripts, as in $Y^{(s)}$, are used to identify the order of the matrix since the algorithm to be presented is defined recursively on n. Since $Y^{(n)}$ is also symmetric, all submatrices Y_i are symmetric. The submatrices are $N_p \times N_p$ complex matrices where N_p is the shortest row or column dimension of the array lattice.

Since the submatrices are of order $N_p \times N_p$, then $Y^{(n)}$ is of order $N_p \cdot (n+1) \times N_p \cdot (n+1)$. A recursive system of equations defines further sets of $N_p \times N_p$ matrices,

$$\left.\begin{aligned}
\psi_o^{(o)} &= Y_o^{-1} Y_1 \\
\Delta^{(-1)} &= Y_o^{-1} \\
\Delta^{(m-1)} &= [I_N - (\psi_{m-1}^{(m-1)})^2]^{-1} \Delta^{(m-2)} \\
\psi_m^{(m)} &= - \Delta^{(m-1)} [\sum_{s=0}^{m-1} \psi_s^{(m-1)T} Y_{m-s} - Y_{m+1}] \\
\psi_r^{(m)} &= \psi_r^{(m-1)} - \psi_{m-r-1}^{(m-1)} \psi_m^{(m)}, \quad 0 \leq r \leq m-1
\end{aligned}\right\} \tag{16}$$

for m=1,2,...,n-1 (superscript T denotes transpose). Then the $N_p \times N_p$ submatrices of $Z = Y^{-1}$, defined by

$$Z = \begin{bmatrix} Z_{00} & Z_{01} & Z_{02} & \cdots & Z_{0n} \\ Z_{10} & Z_{11} & Z_{12} & \cdots & Z_{1n} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ Z_{n0} & Z_{n1} & \cdots & \cdots & Z_{nn} \end{bmatrix} \tag{17}$$

are given by

$$Z_{00} = \Delta^{(n-1)} \tag{18}$$

$$Z_{r0} = - \psi_{r-1}^{(n-1)} \Delta^{(n-1)}, \quad 1 \leq r \leq n \tag{19}$$

$$\begin{aligned} Z_{rs} = Z_{r-1,s-1} &+ \psi_{r-1}^{(n-1)} \Delta^{(n-1)} \psi_{s-1}^{(n-1)T} \\ &- \psi_{n-r}^{(n-1)} \Delta^{(n-1)} \psi_{n-s}^{(n-1)T} \quad 1 \leq r, s \leq n \end{aligned} \tag{20}$$

where $\psi_{-1}^{(n-1)} = -I_{N_p}$, the unit matrix of order N_p and $\psi_r^{(n-1)} = 0$ for $r \geq n$.

The solution for $\vec{V}$, partitioned into subvectors of length N_p denoted V_r $(r=0,1,\ldots,n)$ is

$$V_r = \sum_{s=0}^{n} Z_{rs} I_s \tag{21}$$

where I_s is the sth subvector of I of length N_p. Next define the matrices ϕ_r where

$$\phi_r = \Delta^{(n-1)} \psi_r^{(n-1)T}, \quad r=0,1,\ldots,n-1 \tag{22}$$

and the vectors a_r and b_r where

$$\left.\begin{aligned} a_r &= \sum_{s=0}^{n-r} \phi_{s-1} I_{s+r} \\ b_r &= \sum_{s=0}^{n-r} \phi_{n-s-1} I_{s+r} \end{aligned}\right\} r=0,1,\ldots,n \tag{23}$$

so that the general solution for $r \geq 1$ is

$$V_r = \sum_{s=0}^{r} \psi_{s-1} a_{r-s} - \sum_{s=1}^{r} \psi_{n-s} b_{r-s+1} \,. \tag{24}$$

The algorithm can be briefly summarized by the following steps:

(1) Use the recursion equations (16) to define $\psi_r^{(n-1)}$ and $\Delta^{(n-1)}$.

(2) Form the matrices ϕ_r by equation (22).

(3) Form the vectors a_r and b_r by equation (23).

(4) Form the voltage subvectors, V_r, by equation (24).

Using this algorithm, the number of multiplications and divisions required to invert $Y^{(n)}$ varies as $(n+1)^2 \cdot N_p^3$ with increasing n and N. This should be compared with a variation as $[(n+1) \cdot N_p]^3$ for a general elimination method and demonstrates a considerable improvement when n is large.

Since the computer program was taken from [5], no further description of it will be provided other than mentioning that there are numerous comment cards in LINSLV written using the same notation as the algorithm just described. For the subroutines MATMLT, TRMMLT, MULTTR, and MATVCA, comment cards which describe the matrix multiplication operation performed follow the subroutine statement cards.

In summary, the subroutine LINSLV(CE, V, YS, NP, NW) solves equation (1-15) for $\vec{V}$ given $[Y^{wg} + y^{hs}]$ and $\vec{I}^{imp}$. The arguments of the subroutine statement card represent the following parameters: the first N elements of CE are elements of $\vec{I}^{imp}$, V is the output magnetic current coefficient column matrix, YS is the input matrix $[Y^{wg} + Y^{hs}]$ partitioned in terms of blocks of dimension NP $\times$ NP (only one row of blocks are required $[Y_0, Y_1, \ldots, Y_n]$), NP is the smaller of the row or column dimensions for the array lattice, and NW uses the same definition as NP but replaces the word smaller with larger.

Minimum allocations are given by

```
COMPLEX A(NP*NP), B(NP*NP), C(NP*NP)
```

in subroutines MATMLT, TRMMLT, MULTTR, and MATVCA, and by

```
COMPLEX CE(N+NP*NW+1), PS(2*(NW+1)*NP*NP),
        V(N), YS((NW+1)*NP*NP)
```

in subroutine LINSLV.

```
C       LISTINGS OF THE SUBROUTINES MATMLT.TRMMLT.MULTTR.
C               MATVCA. AND LINSLV
C
        SUBROUTINE MATMLT(A.B.C.NP)
C       CALCULATES C=A*B
        COMPLEX A(4),B(4),C(4),D
        IJ=0
        L=1
        DO 10 I=1,NP
        DO 11 J=1,NP
        IJ=IJ+1
        D=(0.,0.)
        KJ=L
        JK=J
        DO 12 K=1,NP
        D=D+A(JK)*B(KJ)
        JK=JK+NP
        KJ=KJ+1
     12 CONTINUE
        C(IJ)=D
     11 CONTINUE
        L=L+NP
     10 CONTINUE
        RETURN
        END
        SUBROUTINE TRMMLT(A.B.C.NP)
C       CALCULATES C=-TRANSPOSE(A)*B
        COMPLEX A(4),B(4),C(4),D
        IJ=0
        L=1
        DO 10 I=1,NP
        M=1
        DO 11 J=1,NP
        IJ=IJ+1
        D=(0.,0.)
        KI=M
        KJ=L
        DO 12 K=1,NP
        D=D+A(KI)*B(KJ)
        KI=KI+1
        KJ=KJ+1
     12 CONTINUE
        M=M+NP
        C(IJ)=C(IJ)-D
     11 CONTINUE
        L=L+NP
     10 CONTINUE
        RETURN
        END
        SUBROUTINE MULTTR(A.B.C.NP)
C       CALCULATES C=A*TRANSPOSE(B)
        CCMPLEX A(4).B(4).C(4).D
        IJ=0
        DO 10 I=1,NP
        DO 10 J=1,NP
        IJ=IJ+1
        D=(0.,0.)
        IK=I
        JK=J
        DO 11 K=1,NP
```

```
      D=D+A(JK)*B(IK)
      IK=IK+NP
      JK=JK+NP
   11 CONTINUE
      C(IJ)=D
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MATVCA(A,B,C,N,N1)
C     POSITIVE ACCUMULATION OF  A*B IN C IF N1=1 OR NEGATIVE ACCUMULATION
C     A*B IN C IF N1=2
      COMPLEX A(4),B(4),C(4),D
      DO 10 I=1,N
      D=(0.,0.)
      IJ=I
      DO 11 J=1,N
      D=D+A(IJ)*B(J)
      IJ=IJ+N
   11 CONTINUE
      GO TO (12,13),N1
   12 C(I)=C(I)+D
      GO TO 10
   13 C(I)=C(I)-D
   10 CONTINUE
      RETURN
      END
      SUBROUTINE LINSLV(CE,V,YS,NP,NW)
      COMPLEX CE(13),PS(32),V(6),YS(16)
      DIMENSION IA(2)
  100 FORMAT(//10X,'ILLEGAL CALL TO LINSLV - - NW =',I4)
  101 FORMAT(//10X,'ILLEGAL CALL TO LINSLV - - NP =',I4)
      IF(NW.LT.2) GO TO 10
      IF(NP.LT.2) GO TO 11
C     CALC  DEL(-1)  AND  PS((0),0)
      N=NW-1
      NPW=NP*NW
      N2=NP*NP
      DO 12 I=1,N2
      PS(I)=YS(I)
   12 CONTINUE
      CALL LINEQ(PS,NP)
      CALL MATMLT(PS,YS(N2+1),PS(N2+1),NP)
C     IA(1)  =  START ADDRESS IN PS ARRAY OF PS((M-1),0)
C     IA(2)  =  START ADDRESS IN PS ARRAY OF PS((M),0)
C     IA(1)-N2  =  START ADDRESS IN PS ARRAY OF DEL(M-2)
C     IA(2)-N2  =  START ADDRESS IN PS ARRAY OF DEL(M-1)
C     IST+1=2*NW*N2+1  =  START ADDRESS OF AN ADDITIONAL SCRATCH AREA
      IA(1)=N2+1
      IA(2)=NW*N2+N2+1
      IST=2*NW*N2
      MZ=N2+N2+1
      MM=0
C     ITERATE ON M=1,2,...,N.  FOR M=N, ONLY CALC DEL(M-1).
      DO 13 M=1,N
      I0=IA(1)+MM
      MM=MM+N2
      I1=IA(2)+MM
C     I0 IS START ADDRESS OF PS((M-1),M-1)
C     I1 IS START ADDRESS OF PS((M),M)
C     CALC DEL(M-1)
```

```
      CALL MATMLT(PS(IO),PS(IO),PS(IST+1),NP)
      IJ=IST
      DO 14 I=1,NP
      DO 14 J=1,NP
      IJ=IJ+1
      PS(IJ)=-PS(IJ)
      IF(I.EQ.J) PS(IJ)=PS(IJ)+1.
   14 CONTINUE
      CALL LINEQ(PS(IST+1),NP)
      ID=IA(1)-N2
      ID1=IA(2)-N2
      CALL MATMLT(PS(IST+1),PS(ID),PS(ID1),NP)
      IF(M.EQ.N) GO TO 15
C     CALC PS((M),M)
      MZZ=MZ
      MS=IA(1)
      IJ=IST
      DO 16 I=1,N2
      IJ=IJ+1
      PS(IJ)=YS(MZZ)
      MZZ=MZZ+1
   16 CONTINUE
      MZZ=MZ-N2
      DO 17 IS=1,M
      CALL TRMMLT(PS(MS),YS(MZZ),PS(IST+1),NP)
      MS=MS+N2
      MZZ=MZZ-N2
   17 CONTINUE
      MZ=MZ+N2
      CALL MATMLT(PS(ID1),PS(IST+1),PS(I1),NP)
C     CALCULATE PS((M),R) FOR R=0,1,... M-1. (IR=R)
      IOR=IA(1)
      I1R=IA(2)
      IMR=IO
      DO 18 IR=1,M
      CALL MATMLT(PS(IMR),PS(I1),PS(IST+1),NP)
      IMR=IMR-N2
      IJ=IST
      DO 18 I=1,N2
      PS(I1R)=PS(IOR)-PS(IJ+1)
      IJ=IJ+1
      I1R=I1R+1
      IOR=IOR+1
   18 CONTINUE
      I=IA(1)
      IA(1)=IA(2)
      IA(2)=I
   13 CONTINUE
C     HAVE FINISHED ITERATION ON PS.  NOW PUT PHI(R) INTO PS(IA(2))
   15 IPHI=IA(2)-N2
      IPSI=IA(1)
      IOR=IA(1)
      I1R=IA(2)
      DO 19 I=1,N
      CALL MULTTR(PS(IPHI),PS(IOR),PS(I1R),NP)
      IOR=IOR+N2
      I1R=I1R+N2
   19 CONTINUE
C     PUT PHI(-1) IN PS(IPHI)
      J=IPHI
```

```
      DO 20 I=1,N2
      PS(J)=-PS(J)
      J=J+1
   20 CONTINUE
C     NOW HAVE PHI(S),S=-1,0,1,... N-1 STARTING AT PS(IPHI)
C     AND      PSI(S),S= 0,1,2,... N-1 STARTING AT PS(IPSI)
      IB=NW*NP+1
      IC=1
      J=2*NW*NP
      DO 21 I=1,J
      YS(I)=(0.,0.)
   21 CONTINUE
      IV=1
      DO 22 J=1,NW
      NR=NW-J+1
      I1S=IPHI
      I2S=IPHI+N*N2
      IVS=IV
      DO 23 I=1,NR
      CALL MATVCA(PS(I1S),CE(IVS),YS(IC),NP,1)
      CALL MATVCA(PS(I2S),CE(IVS),YS(IB),NP,1)
      I1S=I1S+N2
      I2S=I2S-N2
      IVS=IVS+NP
   23 CONTINUE
      IB=IB+NP
      IC=IC+NP
      IV=IV+NP
   22 CONTINUE
C     NOW CALCULATE V IN I LOCATIONS
      J=NW*NP
      DO 24 I=1,J
      CE(I)=-YS(I)
   24 CONTINUE
      IV=NP+1
      IB=IV+J
      IC=1
      DO 25 IR=1,N
      I1S=IPSI
      I2S=(N-1)*N2+IPSI
      IBS=IB
      ICS=IC
      DO 26 IS=1,IR
      CALL MATVCA(PS(I1S),YS(ICS),CE(IV),NP,1)
      CALL MATVCA(PS(I2S),YS(IBS),CE(IV),NP,2)
      I1S=I1S+N2
      I2S=I2S-N2
      IBS=IBS-NP
      ICS=ICS-NP
   26 CONTINUE
      IB=IB+NP
      IC=IC+NP
      IV=IV+NP
   25 CONTINUE
      DO 27 I=1,NPW
      V(I)=CE(I)
   27 CONTINUE
      RETURN
   10 WRITE(3,100) NW
      RETURN
   11 WRITE(3,101) NP
      RETURN
      END
```

IX. DESCRIPTION OF THE SUBROUTINES LINEQ, DECOMP, AND SOLVE

The subroutine LINEQ(C,LL) is used in subroutine LINSLV to invert a complex matrix. The input to LINEQ consists of a square complex matrix of order LL × LL stored columnwise in C and the dimension variable LL. The output from LINEQ is C^{-1} stored columnwise in C.

Minimum allocations are given by

```
COMPLEX C(LL*LL)
DIMENSION LR(LL).
```

The subroutines DECOMP(N,IPS,UL) and SOLVE(N,IPS,UL,B,X) are called from the main program to solve the matrix equation

$$[Y^{wg} + Y^{hs}]\vec{V} = \vec{I}^{imp} \tag{1-15}$$

when the array lattice is isosceles triangular. This subroutine combination uses the method of Gaussian elimination and LU decomposition described in [6, Section 9]. The input to DECOMP consists of N and the N by N matrix of coefficients $[Y^{wg} + Y^{hs}]$ which is stored by columns in UL. The output from DECOMP is IPS and UL. The output is fed into SOLVE. The rest of the input to SOLVE consists of N and the column of coefficients $\vec{I}^{imp}$ stored in B. Solve puts the solution $\vec{V}$ to the matrix equation in X.

Minimum allocations are given by

```
COMPLEX UL(N*N)
DIMENSION SCL(N), IPS(N)
```

in subroutine DECOMP, and by

```
COMPLEX UL(N*N), B(N), X(N)
DIMENSION IPS(N)
```

in subroutine SOLVE.

DECOMP and SOLVE require roughly $N^3/3$ multiplications and divisions to solve a system of N linear equations whereas using LINEQ in a method which requires an inverse needs roughly $N^3 + N^2$ multiplications and divisions to solve the same system of equations.

```
C     LISTINGS OF THE SUBROUTINES LINEQ,DECOMP,AND SOLVE
C
      SUBROUTINE LINEQ(C,LL)
      COMPLEX C(4),STOR,STO,ST,S
      DIMENSION LR(2)
      DO 10 I=1,LL
      LR(I)=I
   10 CONTINUE
      M1=0
      DO 11 M=1,LL
      K=M
      K2=M1+K
      S1=ABS(REAL(C(K2)))+ABS(AIMAG(C(K2)))
      DO 12 I=M,LL
      K1=M1+I
      S2=ABS(REAL(C(K1)))+ABS(AIMAG(C(K1)))
      IF(S2-S1) 12,12,13
   13 K=I
      S1=S2
   12 CONTINUE
      LS=LR(M)
      LR(M)=LR(K)
      LR(K)=LS
      K2=M1+K
      STOR=C(K2)
      J1=0
      DO 14 J=1,LL
      K1=J1+K
      K2=J1+M
      STO=C(K1)
      C(K1)=C(K2)
      C(K2)=STO/STOR
      J1=J1+LL
   14 CONTINUE
      K1=M1+M
      C(K1)=1./STOR
      DO 15 I=1,LL
      IF(I-M) 16,15,16
   16 K1=M1+I
      ST=C(K1)
      C(K1)=0.
      J1=0
      DO 17 J=1,LL
      K1=J1+I
      K2=J1+M
      C(K1)=C(K1)-C(K2)*ST
      J1=J1+LL
   17 CONTINUE
   15 CONTINUE
      M1=M1+LL
   11 CONTINUE
      J1=0
      DO 18 J=1,LL
      IF(J-LR(J)) 19,20,19
   19 LRJ=LR(J)
      J2=(LRJ-1)*LL
      DO 21 I=1,LL
      K2=J2+I
      K1=J1+I
      S=C(K2)
```

```
      C(K2)=C(K1)
      C(K1)=S
   21 CONTINUE
      LR(J)=LR(LRJ)
      LR(LRJ)=LRJ
      IF(J-LR(J)) 19,20,19
   20 J1=J1+LL
   18 CONTINUE
      RETURN
      END
      SUBROUTINE DECOMP(N,IPS,UL)
      COMPLEX UL(36),PIVOT,EM
      DIMENSION SCL(6),IPS(6)
      DO 5 I=1,N
      IPS(I)=I
      RN=0.
      J1=I
      DO 2 J=1,N
      ULM=ABS(REAL(UL(J1)))+ABS(AIMAG(UL(J1)))
      J1=J1+N
      IF(RN-ULM) 1,2,2
    1 RN=ULM
    2 CONTINUE
      SCL(I)=1./RN
    5 CONTINUE
      NM1=N-1
      K2=0
      DO 17 K=1,NM1
      BIG=0.
      DO 11 I=K,N
      IP=IPS(I)
      IPK=IP+K2
      SIZE=(ABS(REAL(UL(IPK)))+ABS(AIMAG(UL(IPK))))*SCL(IP)
      IF(SIZE-BIG) 11,11,10
   10 BIG=SIZE
      IPV=I
   11 CONTINUE
      IF(IPV-K) 14,15,14
   14 J=IPS(K)
      IPS(K)=IPS(IPV)
      IPS(IPV)=J
   15 KPP=IPS(K)+K2
      PIVOT=UL(KPP)
      KP1=K+1
      DO 16 I=KP1,N
      KP=KPP
      IP=IPS(I)+K2
      EM=-UL(IP)/PIVOT
   18 UL(IP)=-EM
      DO 16 J=KP1,N
      IP=IP+N
      KP=KP+N
      UL(IP)=UL(IP)+EM*UL(KP)
   16 CONTINUE
      K2=K2+N
   17 CONTINUE
      RETURN
      END
      SUBROUTINE SOLVE(N,IPS,UL,B,X)
      COMPLEX UL(36),B(6),X(6),SUM
```

```
      DIMENSION IPS(6)
      NP1=N+1
      IP=IPS(1)
      X(1)=B(IP)
      DO 2 I=2,N
      IP=IPS(I)
      IPB=IP
      IM1=I-1
      SUM=0.
      DO 1 J=1,IM1
      SUM=SUM+UL(IP)*X(J)
    1 IP=IP+N
    2 X(I)=B(IPB)-SUM
      K2=N*(N-1)
      IP=IPS(N)+K2
      X(N)=X(N)/UL(IP)
      DO 4 IBACK=2,N
      I=NP1-IBACK
      K2=K2-N
      IPI=IPS(I)+K2
      IP1=I+1
      SUM=0.
      IP=IPI
      DO 3 J=IP1,N
      IP=IP+N
    3 SUM=SUM+UL(IP)*X(J)
    4 X(I)=(X(I)-SUM)/UL(IPI)
      RETURN
      END
```

X. DESCRIPTION OF THE SUBROUTINE TOPGEN

The subroutine TOPGEN(N, NP, NT, NW, X, Y, T, T1, KR) generates a complete matrix T given one column T1.

Minimum allocations are given by

```
COMPLEX T(N*N), T1(N), T2(N,N)
DIMENSION D(6), X(6), Y(6)
```

The variables in the argument of TOPGEN are defined as follows: N is the dimension of the square matrix T, N_p is the shortest row or column dimension of the array lattice, NT specifies the array lattice type-1 for rectangular and 2 for isosceles triangular, NW is the largest row or column dimension of the array lattice, X(I) and Y(I) are the center coordinates of the Ith aperture and T is the output matrix which is generated from one column T1 of T. KR is a parameter which determines whether or not D(I) is calculated ($D(I) = (X_1-X_I)^2 + (Y_1-Y_I)^2$). If KR = 1, D(I) is evaluated whereas if KR = 2, this step is skipped (D(I) has been determined already by a previous call to TOPGEN).

DO loops 11, 12, 13, and 14 generate [T] given $\vec{T1}$ for a uniformly spaced rectangular array lattice while DO loops 16, 17, 18, 19, 21, and 22 does the same for a uniformly spaced isosceles triangular array lattice. For the rectangular array lattice, [T] which is symmetric block-Toeplitz is determined by forming one column of blocks and generating the rest of [T] using the Toeplitz property. For the isosceles triangular array lattice, the elements T_{ij} below the diagonal of [T] (excluding the elements of the first column which are given (T1)) are determined by comparing the distance squared between the centers of the apertures i and j, D1 (D1 of $T_{ij} = (x_j-x_i)^2 + (y_j-y_i)^2$), to the distance values associated with the elements of the first column, D (D of $T_{i1} = (x_1-x_i)^2 + (y_1-y_i)^2$). When D1 = D, the value used for T_{ij} which has an associated D1 value is the same as the corresponding element of the first column with an associated D value. The elements above the diagonal are determined from symmetry, $y_{ij} = y_{ji}$ (consequence of Galerkin's method).

```
      SUBROUTINE TOPGEN(N,NP,NT,NW,X,Y,T,T1,KR)
      COMPLEX T(36),T1(6),T2(6,6)
      DIMENSION D(6),X(6),Y(6)
      NPW=NP*NW
      NPWP=NPW*NP
      N1=N-1
      IF(NT.EQ.2) GO TO 10
      DO 11 I=1,NW
      L1=1
      DO 12 J=1,NP
      DO 12 K=1,NP
      L=1+IABS(K-J)+NP*(I-1)
      T2(I,L1)=T1(L)
      L1=L1+1
   12 CONTINUE
   11 CONTINUE
      DO 13 I=1,NW
      DO 13 I1=1,NW
      IF(I.EQ.1) II=I1
      IF(I.GT.1) II=I-I1+1
      IF((I.GT.1).AND.(II.LE.0)) II=I1-I+1
      I2=1
      DO 14 J=1,NP
      DO 14 K=1,NP
      L1=(I-1)*NPWP+(I1-1)*NP+(J-1)*NPW+K
      T(L1)=T2(II,I2)
      I2=I2+1
   14 CONTINUE
   13 CONTINUE
      RETURN
   10 IF(KR.EQ.2) GO TO 15
      DO 16 I=1,N
      D(I)=(X(I)-X(1))*(X(I)-X(1))+(Y(I)-Y(1))*(Y(I)-Y(1))
   16 CONTINUE
   15 I1=1
      K1=1
      DO 17 I=1,N1
      DO 18 J=I1,N
      D1=(X(J)-X(I))*(X(J)-X(I))+(Y(J)-Y(I))*(Y(J)-Y(I))
      DO 19 K=1,N
      IF(ABS(D1-D(K)).LT.0.01) T(K1)=T1(K)
      IF(ABS(D1-D(K)).LT.0.01) GO TO 20
   19 CONTINUE
   20 K1=K1+1
   18 CONTINUE
      IF(I.EQ.1) I1=I1+1
      I1=I1+1
      K1=K1+I1-1
   17 CONTINUE
      J2=0
      DO 21 J=1,N
      J1=J
      DO 22 I=1,J
      J3=J2+I
      IF(I-J) 23,24,23
   24 T(J1)=T1(1)
      GO TO 22
   23 T(J3)=T(J1)
      J1=J1+N
   22 CONTINUE
      J2=J2+N
   21 CONTINUE
      RETURN
      END
```

XI. DESCRIPTION OF THE MAIN PROGRAM WITH SAMPLE INPUT-OUTPUT DATA

The main program computes the x,y coordinates of the centers of the apertures in the array, the complex coefficients V_i which determine the magnetic currents $\underset{\sim}{M}^i$ according to (1-6) and the scattering coefficients S_{ij} according to (1-103) and (1-104). The main program calls the subroutines AY, YHSP, TOPGEN, CSMTZ, DECOMP, SOLVE, and LINSLV.

The data cards are read into the main program according to

```
        READ (1,100) N, AL, AL1, BL, BL1, ER, LM, LN
    100 FORMAT (I4, 5F7.4, 2I3)
        READ (1,102) NC, NR, NT, NE, DX, DY, DT
    102 FORMAT (4I4, 3F7.4).
```

The variables AL, AL1, BL, and BL1 are respectively a/λ, a'/λ, b/λ, and b'/λ where λ is the free space wavelength. ER is the relative dielectric constant ε_r of (1-34) and (1-35) inside the waveguide. The variables LM and LN are respectively the total number of m and n modes used in determining A_{ik}^{TE} (1-30) and A_{ik}^{TM} (1-31) (only odd m starting with m=1 and even n starting with n=0 (TE) or n=2 (TM) are considered due to the $\sin\frac{m\pi}{2}\cos\frac{n\pi}{2}$ factor appearing in (1-33)). LM which represents the contribution of the mth waveguide mode to Y^{wg} should be chosen so that the contribution of the $(1/(m^2/a^2 - 1/a'^2))\cos(m\pi a'/2a)$ factor in (1-33) results in very small A_{ik}^{TE} and A_{ik}^{TM} values. LN which represents the contribution of the nth mode to Y^{wg} should be chosen so that the argument $n\pi b'/2b$ of the sin( )/( ) factor in (1-33) is greater than π. NC is the number of columns of apertures measured in the x direction. NR is the number of rows of apertures measured in the y direction. NT specifies the array lattice type, 1 for rectangular and 2 for isosceles triangular. NE is the driven aperture number. DX is the distance per unit wavelength between the closest outer waveguide edges in the x direction while DY is the same but in the y direction (see Fig. 1-1). DT is the waveguide wall thickness per unit wavelength.

Minimum allocations are given by

```
COMPLEX CE(N+NP*NW+1), S(N*N), S3(N),
    V(N), YO(2*LM*LN+LM), YS(N*N),
    YS1((NW+1)*NP*NP), YSP(N)
DIMENSION A(2*LM*LN), IPS(N), X(N), Y(N)
```

in the main program, by

```
COMPLEX Y(2*LM*LN + LM)
DIMENSION A(2*LM*LN), F(LM), STE(LN), STM(LN)
```

in the subroutine AY, by

```
COMPLEX YSP(N)
DIMENSION X(N), Y(N)
```

in the subroutine YHSP, by

```
COMPLEX T(N*N), T1(N), T2(N,N)
DIMENSION D(N), X(N), Y(N)
```

in the subroutine TOPGEN, by

```
COMPLEX A(N), B(N), E(N), ES(N-1), X(N)
```

in the subroutine CSMTZ, by

```
COMPLEX UL(N*N)
DIMENSION SCL(N), IPS(N)
```

in the subroutine DECOMP, by

```
COMPLEX UL(N*N), B(N), X(N)
DIMENSION IPS(N)
```

in the subroutine SOLVE, and by

```
COMPLEX CE(N+NP*NW+1), PS(2*(NW+1)*NP*NP),
        V(N), YS((NW+1)*NP*NP)
```

in the subroutine LINSLV where NP is the smaller of the number of rows or columns of apertures for the rectangular array lattice while NW uses the same definition but replaces the word smaller by larger.

DO loop 11 calculates the x and y coordinates of the centers of the apertures if the array has a rectangular lattice while DO loop 15 does the same if the lattice is isosceles triangular. Referring to equations (1-30) to (1-35), statement 40 stores

A_k^{TE} in A(k = (m+1)/2 + n/2 * LM)

Y_k^{TE} in YO(k = (m+1)/2 + n/2 * LM)

m = 1,3,5,..., LM

n = 0,2,4,..., LN

A_k^{TM} in A(k = LM*LN + (m+1)/2 + (n-2)/2 * LM)

Y_k^{TM} = YO(k = LM*LN + (m+1)/2 + (n-2)/2 * LM)

m = 1,3,5,..., LM

n = 2,4,6,..., LN .

Note that only odd m and even n modes are calculated and that the first subscript of A_{ik}^{TE} and A_{ik}^{TM} in (1-30) and (1-31) has been dropped since the same expansion function is used in every waveguide region - $A_{1k} = A_{2k} = \ldots = A_{Nk} = A_k$. DO loop 18 calculates Y^{wg}. Statement 41 stores the first column of Y^{hs} in YSP. Statement 42 generates the complete Y^{hs} matrix given the first column and stores it in YS by columns. DO loop 19 and statement 43 generate $\vec{I}^{imp}$ defined by (1-106) and (1-107) and store it in CE.

If the array is linear, DO loop 21 is used to calculate $[Y^{wg} + Y^{hs}]$ and statement 44 calculates the coefficient vector $\vec{V}$ defined by (1-15) and stores it in V.

If the array contains at least two rows and columns and has a rectangular lattice, statement 20 and DO loops 24, 25 calculate the symmetric submatrix blocks of $[Y^{wg} + Y^{hs}]$ given $[Y^{wg}]$ and a column of $[Y^{hs}]$. If the array contains at least two rows and columns and has an isosceles triangular lattice, DO loop 27 calculates $[Y^{wg} + Y^{hs}]$ given $[Y^{wg}]$ and $[Y^{hs}]$. Once $[Y^{wg} + Y^{hs}]$ has been determined for these two cases, statement 26 stores $\vec{V}$ in V for the rectangular array while statements 45 and 46 store $\vec{V}$ in V for the isosceles triangular array.

DO loop 31 calculates one column of the scattering matrix defined by (1-103) and (1-104) and stores it in S3. Statement 47 generates the complete scattering matrix [S] given the first column and stores it in S.

The following is a listing of the main program with sample input-output data. Figure 2 shows the coupled power (20 log $|S_{i1}|$) and phase of S_{i1} (i = 2,6) between a driven element and the rest of the elements for the 2 × 3 waveguide-fed rectangular aperture array example used in the sample input-output data.

```
C      LISTING OF THE MAIN PROGRAM AND SAMPLE DATA
C
//XXXX WATFIV  (XXXX,XX,1,2),'XXXX',REGION=250K
$JOB          XXXX,TIME=1,PAGES=30
C
C      MAIN PROGRAM
C      THIS PROGRAM CALLS THE SUBROUTINES AY,YHSP,TOPGEN,CSMTZ,
C           DECOMP,SOLVE, AND LINSLV
       COMPLEX CE(13),G3,G5,G7,G9,S(36),S3(6),U,V(6)
       COMPLEX YO(55),YS(36),YS1(16),YSP(6),YWG
       DIMENSION A(50),IPS(6),X(6),Y(6)
       COMMCN AL1,BL1,PI,PI2,PI3,U/R1/ETA
       COMMON /R3/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11
       COMMON /R5/AL,BL/R6/ER,LM,LN,NTS
   100 FORMAT(I4,5F7.4,2I3)
   101 FORMAT(/5X,'N',5X,'AL',5X,'AL1',4X,'BL',5X,'BL1',4X,'ER',5X,'LM',
      13X,'LN'/2X,I4,2X,5F7.4,2X,I3,2X,I3)
   102 FORMAT(4I4,3F7.4)
   103 FORMAT(//4X,'NC',5X,'NR',5X,'NT',5X,'NE',5X,'DX',5X,'DY',5X,'DT'/
      12X,I4,3X,I4,3X,I4,3X,I4,2X,3F7.4)
   104 FORMAT(/4X,'X'/(2X,5E14.7))
   105 FORMAT(/4X,'Y'/(2X,5E14.7))
   106 FORMAT(/4X,'Y-WG'/2X,2E14.7)
   107 FORMAT(/4X,'Y-HALF SPACE'/(2X,5E14.7))
   108 FORMAT(/4X,'V - - UNKNOWN VECTOR'/(2X,5E14.7))
   109 FORMAT(/4X,'S - - SCATTERING MATRIX'/(2X,5E14.7))
       READ(1,100) N,AL,AL1,BL,BL1,ER,LM,LN
       WRITE(3,101) N,AL,AL1,BL,BL1,ER,LM,LN
       READ(1,102) NC,NR,NT,NE,DX,DY,DT
       WRITE(3,103) NC,NR,NT,NE,DX,DY,DT
       PI=3.141593
       PI2=2.*PI
       PI3=PI/2.
       ETA=376.730
       U=(0.,1.)
       N1=N+1
       N2=LM*LN+1
       NN=N*N
       AL2=AL/2.+DT
       BL2=BL/2.+DT
       DX2=DX/2.
```

```
      NP=AMINO(NC,NR)
      NW=AMAXO(NC,NR)
      NTS=2*LM*LN
      NP2=NP*NP
      G1=AL1/PI
      G2=AL1/PI2
      G3=-U*G2
      G4=G1*G1
      G5=-U*G4
      G6=G1*G2
      G7=-U*G6
      G8=G1*G4
      G9=-U*G8
      G10=(1.-0.25/(AL1*AL1))
      G11=(1.+0.25/(AL1*AL1))*G1
      S1=DX+2.*DT+AL
      S2=DY+2.*DT+BL
      IF(NT.EQ.2) GO TO 10
      K=1
      DO 11 I=1,NW
      DO 11 J=1,NP
      IF(NC.GT.NR) GO TO 12
      X(K)=AL2+(J-1)*S1
      Y(K)=BL2+(I-1)*S2
      GO TO 13
   12 X(K)=AL2+(I-1)*S1
      Y(K)=BL2+(J-1)*S2
   13 K=K+1
   11 CONTINUE
      GO TO 14
   10 K=1
      DO 15 I=1,NR
      K1=2-(I-2*(I/2))
      DO 15 J=1,NC
      X(K)=(1+K1/2)*AL2+(J-1)*S1+K1/2*DX2
      Y(K)=BL2+(I-1)*S2
      K=K+1
   15 CONTINUE
   14 WRITE(3,104) (X(I),I=1,N)
      WRITE(3,105) (Y(I),I=1,N)
   40 CALL AY(A,YO)
      YWG=(0.,0.)
      K=1
      DO 18 I=1,NTS
      IF(I.EQ.N2) K=K+LM
      YWG=YWG+A(I)*YO(K)*A(I)
      K=K+1
   18 CONTINUE
      WRITE(3,106) YWG
   41 CALL YHSP(N,X,Y,YSP)
      KR=1
   42 CALL TOPGEN(N,NP,NT,NW,X,Y,YS,YSP,KR)
      WRITE(3,107) (YS(I),I=1,NN)
      DO 19 I=1,N
      CE(I)=(0.,0.)
   19 CONTINUE
   43 CE(NE)=2.*YO(1)*A(1)
      IF((NC.GE.2).AND.(NR.GE.2)) GO TO 20
      YSP(1)=YSP(1)+YWG
```

```
      DO 21 I=1,N
      CE(I)=CE(I)/YSP(1)
      IF(I.EQ.1) GO TO 21
      YSP(I)=YSP(I)/YSP(1)
   21 CONTINUE
   44 CALL CSMTZ(N,YSP,CE,V)
      GO TO 22
   20 YSP(1)=YSP(1)+YWG
      IF(NT.EQ.2) GO TO 23
      DO 24 I=1,NW
      DO 25 J=1,NP
      DO 25 K=1,NP
      L=(I-1)*NP2+(J-1)*NP+K
      L1=(I-1)*NP+IABS(K-J)+1
      YS1(L)=YSP(L1)
   25 CONTINUE
   24 CONTINUE
   26 CALL LINSLV(CE,V,YS1,NP,NW)
      GO TO 22
   23 J=-N
      DO 27 I=1,N
      J=J+N1
      YS(J)=YS(J)+YWG
   27 CONTINUE
   45 CALL DECOMP(N,IPS,YS)
   46 CALL SOLVE(N,IPS,YS,CE,V)
   22 WRITE(3,108) (V(I),I=1,N)
      DO 31 I=1,N
      S3(I)=V(I)*A(1)
      IF(I.EQ.NE) S3(I)=S3(I)-1.
   31 CONTINUE
      KR=2
   47 CALL TOPGEN(N,NP,NT,NW,X,Y,S,S3,KR)
      WRITE(3,109) (S(K),K=1,NN)
      STOP
      END
$DATA
   6 1.0000 0.6500 0.4761 0.3095 1.0000   5   5
   3    2    1    1 0.9000 0.2000 0.0515
$STOP
//
```

PRINTED OUTPUT

N	AL	AL1	BL	BL1	ER	LM	LN
6	1.0000	0.6500	0.4761	0.3095	1.0000	5	5

NC	NR	NT	NE	DX	DY	DT
3	2	1	1	0.9000	0.2000	0.0515

```
 X
0.5515000E+00 0.5515000E+00 0.2554499E+01 0.2554499E+01 0.4557498E+01
0.4557498E+01

 Y
0.2895499E+00 0.1068649E+01 0.2895499E+00 0.1068649E+01 0.2895499E+00
0.1068649E+01

 Y-WG
0.1289030E-02-0.8829894E-04
```

Y-HALF SPACE

```
 0.1375110E-02 0.6628551E-03-0.3466108E-03 0.1446004E-03-0.1527445E-04
-0.2117092E-05 0.8159508E-06 0.1840072E-04-0.3747291E-05-0.1479488E-06
-0.2525312E-05 0.3105473E-05-0.3466108E-03 0.1446004E-03 0.1375110E-02
 0.6628551E-03 0.8159508E-06 0.1840072E-04-0.1527445E-04-0.2117092E-05
-0.2525312E-05 0.3105473E-05-0.3747291E-05-0.1479488E-06-0.1527445E-04
-0.2117092E-05 0.8159508E-06 0.1840072E-04 0.1375110E-02 0.6628551E-03
-0.3466108E-03 0.1446004E-03-0.1527445E-04-0.2117092E-05 0.8159508E-06
 0.1840072E-04 0.8159508E-06 0.1840072E-04-0.1527445E-04-0.2117092E-05
-0.3466108E-03 0.1446004E-03 0.1375110E-02 0.6628551E-03 0.8159508E-06
 0.1840072E-04-0.1527445E-04-0.2117092E-05-0.3747291E-05-0.1479488E-06
-0.2525312E-05 0.3105473E-05-0.1527445E-04-0.2117092E-05 0.8159508E-06
 0.1840072E-04 0.1375110E-02 0.6628551E-03-0.3466108E-03 0.1446004E-03
-0.2525312E-05 0.3105473E-05-0.3747291E-05-0.1479488E-06 0.8159508E-06
 0.1840072E-04-0.1527445E-04-0.2117092E-05-0.3466108E-03 0.1446004E-03
 0.1375110E-02 0.6628551E-03
```

V - - UNKNOWN VECTOR

```
-0.1237808E+01 0.2902574E+00-0.1171501E+00 0.1303046E+00-0.4720550E-02
 0.3497620E-02 0.3164612E-02 0.9165186E-02-0.1368488E-02 0.1222063E-02
-0.5777453E-03 0.2339229E-02
```

S - - SCATTERING MATRIX

```
-0.7309630E-01-0.2173524E+00 0.8772510E-01-0.9757555E-01 0.3534873E-02
-0.2619111E-02-0.2369745E-02-0.6863132E-02 0.1024760E-02-0.9151131E-03
 0.4326310E-03-0.1751676E-02 0.8772510E-01-0.9757555E-01-0.7309630E-01
-0.2173524E+00-0.2369745E-02-0.6863132E-02 0.3534873E-02-0.2619111E-02
 0.4326310E-03-0.1751676E-02 0.1024760E-02-0.9151131E-03 0.3534873E-02
-0.2619111E-02-0.2369745E-02-0.6863132E-02-0.7309630E-01-0.2173524E+00
 0.8772510E-01-0.9757555E-01 0.3534873E-02-0.2619111E-02-0.2369745E-02
-0.6863132E-02-0.2369745E-02-0.6863132E-02 0.3534873E-02-0.2619111E-02
 0.8772510E-01-0.9757555E-01-0.7309630E-01-0.2173524E+00-0.2369745E-02
-0.6863132E-02 0.3534873E-02-0.2619111E-02 0.1024760E-02-0.9151131E-03
 0.4326310E-03-0.1751676E-02 0.3534873E-02-0.2619111E-02-0.2369745E-02
-0.6863132E-02-0.7309630E-01-0.2173524E+00 0.8772510E-01-0.9757555E-01
 0.4326310E-03-0.1751676E-02 0.1024760E-02-0.9151131E-03-0.2369745E-02
-0.6863132E-02 0.3534873E-02-0.2619111E-02 0.8772510E-01-0.9757555E-01
-0.7309630E-01-0.2173524E+00
```

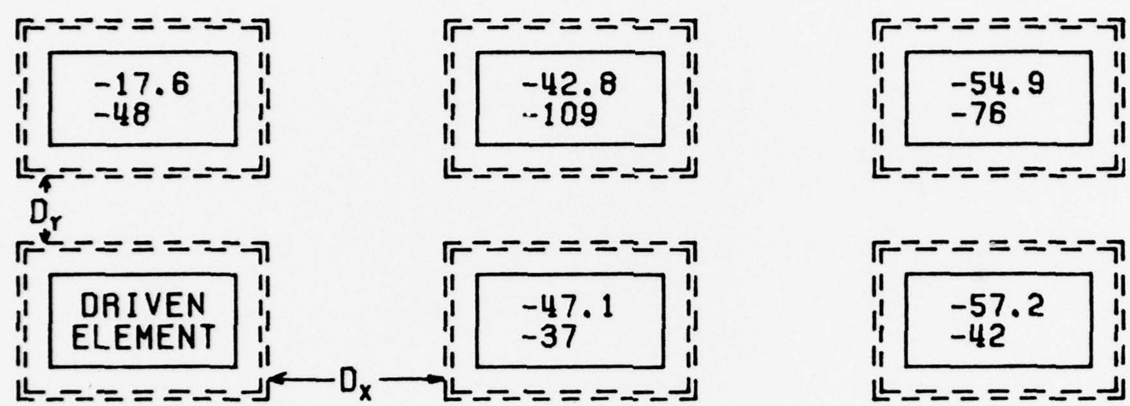


Fig. 2. The coupled power (20 log $|S_{i1}|$) and phase of S_{i1} (i = 2,6) for a 6 element waveguide-fed aperture array with a rectangular lattice where $a/\lambda = 1.0000$, $a'/\lambda = 0.6500$, $b/\lambda = 0.4761$, $b'/\lambda = 0.3095$, $D_x/\lambda = 0.9000$, $D_y/\lambda = 0.2000$ and $D_t/\lambda = 0.0515$. The upper number in each aperture represents coupled power in dB while the lower number represents the phase of S_{i1} in degrees.

REFERENCES

[1] J. Luzwick and R. F. Harrington, "Mutual Coupling in a Finite Planar Rectangular Waveguide Antenna Array," Tech. Rept. No. 7, Contract No. N00014-76-C-0225, Office of Naval Research, June 1978.

[2] V. I. Krylov, Approximate Calculation of Integrals, translated by A. H. Stroud, Macmillan Co., New York, 1962.

[3] Shalhav Zohar, "Toeplitz Matrix Inversion: The Algorithm of W. F. Trench," J. Assoc. Comp. Mach., vol. 16, No. 4, October 1969, pp. 592-601.

[4] Shalhav Zohar, "The Solution of a Toeplitz Set of Linear Equations," J. Assoc. Comp. Mach., vol. 21, No. 2, April 1974, pp. 272-276.

[5] D. H. Sinnott, "An Improved Algorithm for Matrix Analysis of Linear Antenna Arrays," WRE-Technical Note - 1066 (AP), Weapons Research Establishment, Dept. of Defense, Salisburg, South Australia.

[6] G. E. Forsythe and C. B. Moler, Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967, Section 9.

Jan 1974

DISTRIBUTION LIST FOR ONR ELECTRONICS PROGRAM OFFICE

Director
Advanced Research Projects Agency
Attn: Technical Library
1400 Wilson Boulevard
Arlington, Virginia 22209

Office of Naval Research
Electronics Program Office (Code 427)
800 North Quincy Street
Arlington, Virginia 22217

Office of Naval Research
Code 105
800 North Quincy Street
Arlington, Virginia 22217

Naval Research Laboratory
Department of the Navy
Attn: Code 2627
Washington, D. C. 20375

Office of the Director of Defense
Research and Engineering
Information Office Library Branch
The Pentagon
Washington, D. C. 20301

U. S. Army Research Office
Box CM, Duke Station
Durham, North Carolina 27706

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

Director National Bureau of Standards
Attn: Technical Library
Washington, D. C. 20234

Commanding Officer
Office of Naval Research Branch Office
536 South Clark Street
Chicago, Illinois 60605

San Francisco Area Office
Office of Naval Research
50 Fell Street
San Francisco, California 94102

Air Force Office of Scientific Research
Department of the Air Force
Washington, D. C. 20333

Commanding Officer
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

Commanding Officer
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210

Director
U. S. Army Engineering Research
and Development Laboratories
Fort Belvoir, Virginia 22060
Attn: Technical Documents Center

ODDR&E Advisory Group on Electron Devices
201 Varick Street
New York, New York 10014

New York Area Office
Office of Naval Research
207 West 24th Street
New York, New York 10011

Air Force Weapons Laboratory
Technical Library
Kirtland Air Force Base
Albuquerque, New Mexico 87117

Air Force Avionics Laboratory
Air Force Systems Command
Technical Library
Wright-Patterson Air Force Base
Dayton, Ohio 45433

Air Force Cambridge Research Laboratory
L. G. Hanscom Field
Technical Library
Cambridge, Massachusetts 02138

Harry Diamond Laboratories
Technical Library
Connecticut Avenue at Van Ness, N. W.
Washington, D. C. 20438

Naval Air Development Center
Attn: Technical Library
Johnsville
Warminster, Pennsylvania 18974

Naval Weapons Center
Technical Library (Code 753)
China Lake, California 93555

Naval Training Device Center
Technical Library
Orlando, Florida 22813

Naval Research Laboratory
Underwater Sound Reference Division
Technical Library
P. O. Box 8337
Orlando, Florida 32806

Navy Underwater Sound Laboratory
Technical Library
Fort Trumbull
New London, Connecticut 06320

Commandant, Marine Corps
Scientific Advisor (Code AX)
Washington, D. C. 20380

Naval Ordnance Station
Technical Library
Indian Head, Maryland 20640

Naval Ship Engineering Center
Philadelphia Division
Technical Library
Philadelphia, Pennsylvania 19112

Naval Postgraduate School
Technical Library (Code 0212)
Monterey, California 93940

Naval Missile Center
Technical Library (Code 5632.2)
Point Mugu, California 93010

Naval Ordnance Station
Technical Library
Louisville, Kentucky 40214

Naval Oceanographic Office
Technical Library (Code 1640)
Suitland, Maryland 20390

Naval Explosive Ordnance Disposal Facility
Technical Library
Indian Head, Maryland 20640

Naval Electronics Laboratory Center
Technical Library
San Diego, California 92152

Naval Undersea Warfare Center
Technical Library
3202 East Foothill Boulevard
Pasadena, California 91107

Naval Weapons Laboratory
Technical Library
Dahlgren, Virginia 22448

Naval Ship Research and Development Center
Central Library (Code L42 and L43)
Washington, D. C. 20007

Naval Ordnance Laboratory White Oak
Technical Library
Silver Spring, Maryland 20910

Naval Avionics Facility
Technical Library
Indianapolis, Indiana 46218